DOCUMENT RESUME

ED 253 198 IR 011 474

AUTHOR Tilmann, Martha J.
TITLE Artificial Intelligence and Its Importance in Education.
PUB DATE 84
NOTE 33p.; Independent study paper, Michigan State University.
PUB TYPE Information Analyses (070) -- Viewpoints (120) -- Reports - Descriptive (141)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS *Artificial Intelligence; Clinical Diagnosis; Cognitive Processes; *Computer Assisted Instruction; *Computers; *Computer Software; Decision Making; Problem Solving; *Robotics
IDENTIFIERS Computer Uses in Education; *Expert Systems

ABSTRACT

Artificial intelligence, or the study of ideas that enable computers to be intelligent, is discussed in terms of what it is, what it has done, what it can do, and how it may affect the teaching of tomorrow. An extensive overview of artificial intelligence examines its goals and applications and types of artificial intelligence including (1) expert systems, e.g., MYCIN, a rule-based system program that can diagnose infectious diseases and recommend treatment, and XCON, an expert system that synthesizes information; (2) language understanding, e.g., the program ELIZA which carries on a conversation with the user much as a psychiatrist would; (3) image understanding and robotics, e.g., CONSIGHT, a program that uses binary images to locate objects in a conveyor belt; and (4) learning, i.e., the computer's ability to learn. Specific artificial intelligence developments in the field of education are described, including the program WUSOR, which was written to advise users of the game WUMPUS, a maze-exploration program that encourages deductive processes and develops a sense of probabilities. The final summary reviews the good qualities of computer assisted instruction programs and suggests ways in which artificial intelligence could aid or be incorporated in their development. Twelve references and twelve items of supporting literature are listed. (LMM)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

IR

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

ED253198

Artificial Intelligence and Its Importance In Education.

Fall 1984

Martha J.Tilmann
Independent Study under John Vinsonhaler Ph.D.
Michigan State University
East Lansing, MI 48823
Copyright 1984

IR011474

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Martha J. Tilmann

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

INTRODUCTION

Most indiviuals, even well educated ones, draw a blank when confronted with the concept of Artificial Intelligence (A.I.). It is a relatively new field, the original work having been started in the 1950's. A.I. has been given varying degrees of importance by goverment and other researchers, but is currently on an upswing and is considered one of the most important areas of study in computer science. In education its usage is on the increase. A.I. can be used as a tool to investigate critical questions in the philosophy of education; i.e. the computer can be used as a model in order to study the learning process. A.I. can also be used to give indiviualized instruction with immediate meaningfull feedback. For these reasons it is vital that educators and school administrators become familiar with what A.I. is, what it has done, what it can do, and how it may affect the teaching of tomorrow.

PART I. AN EXTENSIVE OVERVIEW OF WHAT ARTIFICIAL INTELLIGENCE IS

WHAT IS A.I.?

Artificial Intelligence is a field of study that includes subcategories such as "Language Understanding," "Image Understanding," "Learning," "Expert Systems," "Goal reduction," "Constraint Exploitaion". Most of these will be covered in some detail later. For now A.I. can simply be thought of as "the study of ideas that enable computers to be intelligent" (1). The challenge is to define intelligence. Is it the capacity to

apprehend facts and to reason about them? Is it the ability to speak and understand language? Is it the ability to manipulate objects to be used as tools? Intelligence is, of course, all of these things and more.

WHAT ARE A.I. GOALS?

Winston, in his book on A.I. (1) says the goals of the field of A.I. can be clearly defined by two central themes. One is to make computers more useful, and the other is to understand the principles that make intelligence possible. Accomplishing the latter goal will aid studies in psychology as well as in computer science. A.I. specialists have often studied human information processing to help them make computers intelligent. In turn, theories derived from computer science can be used in the development of of new educational methods. In field of education, accomplishing the first goal could provide a tool that not only reacts to students' mistakes but *understands* them. Computers could go beyond the text book and offer students visual displays of orbiting planets or, say, aural renditions of 16th century madrigals.

To better understand what A.I. is, it will be helpful to understand four of the major areas of its application. These areas include, "Expert Systems," "Language Understsanding," "Image Understanding and Robotics," and "Learning". The specific methods employed to implement these programs will be briefly explained in later sectons.

EXPERT SYSTEMS

An *expert system* is a problem-solving computer program that can reach a level of performance comparable to that of a human expert in some specialized problem domain. These programs are also called *rule-based systems* or *production systems* . Some *rule-based systems* do synthesis. Other *expert systems* do analysis.

A famous expert system that synthesies information is a program called XCON. Originally called R1, it was developed cooperatively by John McDermott at Carnegie-Mellon University (2), Arnold Kraft and Dennis O'Connor at the Digital Equipment, and their associates. XCON's expertise is in the organization of computer science components. When a company buys a computer system, there are many components: a central processor, memory, terminals, tape drives, disk drives, various peripheal controllers and other paraphernalia. All of these must be arranged along proper input/output lines, each electrical module must be placed in an appropriate cabinet, in a suitable slot of a suitable back plane. This process is called configuration, and can lead to hundreds of alternative arrangements. XCON handles this tedious job with endless patience and great efficiency. EXON's data structures and control units will be discussed later.

MYCIN, written by Edward Shortliffe in 1975 (3), is a program that can diagnose infectious diseases and recommend treatment. It is a rule based system that does analysis. MYCIN interacts with the user, asks questions about symptoms of the disease, and then

determines a diagnosis. At each point, the question MYCIN asks depends on the current hypothesis and the answers to previous questions. The initial questions are similar to a checklist; they then vary as evidence builds. A session with MYCIN may proceed as follows.

- What is the patient's name?
  John Doe
- Male or female?
  Male.
- Age?
  He is 55.
- Have you obtained positive cultures indicating general type?
  Yes.
- What type of infection is it?
  Primary Bacteremia.
- Let's call the most recent positive culture CULTURE-1. From what site was CULTURE-1 taken?
  From the blood.
- When?
  May 9.
- Let's call the first significant organism from this blood culture ORGANISM-1. Do you know the identity of ORGANISM-1?
  No.
- Have there been positive cultures from sites that are not associated with infections about which you are seeking advise?
  No.
- Did ORGANISM-1 grow aerobically?
  Yes.
- Did ORGANISM-1 grow anaerobically?
  Yes.
- What is the suspected portal of entry of ORGANISM-1?
  The gastrointestinal tract.
  .
  .
  .

When MYCIN feels it has a grasp on what the assimilated facts suggest, it responds with some additional short line questions to verify and/or determine the probabilities of posssible diognoses and then recomments a treatment.

- My recommendation will be based on the opinion that the identity of ORGANISM-1 may be
    1) Pseudomonas-aeruginosa
    2) Klebsiella-pneumoniae
    3) E. Coli
    4) Bacteroides-fragilis
    5) Enterobacter
    6) Proteus-non-mirabilis

  On a scale of 0 to 4, where higher numbers indicate greater severity, how would you rate the patient's degree of sickness?
    3
- Does the patient have a clinically significant allergic reation to any antimicrobial agent?
    No.
- How much does the patient weigh?
    70 kilograms.
- My first recommendation is as follows: In order to cover for items 1, 2, 3, 5, and 6, give gentamycin using a dose of 119 mg (1.7 mg/kg) q8h IV [or IM] for 10 days. Modify dose in renal failure. And in order to cover for item 4, give clindamycin using a dose of 595 mg . . .

In 1977 MYCIN made diagnoses in agreement with 72 percent the medical experts. This figure has probably improved since then.

Above I have described two famous expert systems. What makes them different from traditional programming will now be exlained. Ordinary computer programs organize knowledge on two levels: data and program. Most expert systems, however, organize on three levels: data, knowledge, and control. On the data level, knowledge about the particular problem is collected and a record of the current state of affairs is kept. Data structures used to represent this information may include formulas in first-order predicate logic, "frames", or semantic networks (see reference (4) for details on these structures).

The knowledge level is the problem-solving level. It contains procedural intructions that manipulate the data in order

to solve the problem. Several different problem solving techniques have been developed. The most widely used are pattern invoked programs, specifically production rules. Hence the names *rule-based system* and *production systems* . A rule has the form:

IF condition THEN primitive action

The "condition" is usually a list of declaratives that test properties about the current state. The "primative action" is some simple action that changes the current state. The following are examples.

```
From XCON:
  IF  the context is layout and assigning a power supply
      an sbi module of any type has been put in a cabinet
      the position it occupies in the cabinet is known
      there is space available in the cabinet for power supply
      there is no available power supply
      the voltage and frequency of the components is known
  THEN   add an appropriate power supply
```

```
From MYCIN:
  IF (1) the infection is primary-bacteremia
     (2) the site of the culture is one of the sterile sites
     (3) the suspected portal of entry of the organism is the
         gastrointestinal tract,
  THEN  there is suggestive evidence that the identity of the
        organism is bacteroides
```

XCON and MYCIN have 2500 and 500 of these rules respectively.

Level one, the data, and level two, the knowledge, interact with and are directed by level three, the control. There are a number of possible strategies designed to perform this task, i.e. state-space search (1), graph-searching (5), propagation of constraints (6,7), and problem reduction (8,9) are some strategies. Each of these strategies has many specific techniques

within it that has been developed by A.I. researchers to create efficient and fast processing of masses of information.

Together these three levels make computer expert systems a unique and powerful tools. However, there are difficulties in creating expert systems and in having them utilized: 1) cost, 2) time it requires to acquire knowledge from experts, 3) software limitations, and 4) lack of user friendliness. Thus, expert systems are not yet widely used.

LANGUAGE UNDERSTANDING

Teaching computers how to understand natural language---English or French as opposed to Fortran or Pascal---has been an interest of developers ever since the discipline of A.I. was started nearly 40 years ago. Initially work was directed toward simple word to word translation. Language encodement was seen as a matter of words, definitions and vocabulary. Legend has it that an early researcher asked his computer to translate "the spirit is willing, but the flesh is weak," first into Russian then back to English. The result was "the vodka is good, but the meat is rotten". There was a growing dissatisfaction with this narrow approach. There have been continued attempts to further understand how we as humans construct language, so we in turn might better understand how this process could be programmed into a computer. In order for a computer to *understand* language it would need to have an enormous body of shared knowledge about the world, an acute sensitivity to subtle differences and context, and

an intuitive insight into human goals and beliefs.

One of the most famous early works in this area is ELIZA, written by Joseph Weizenbaum of the Massachusetts Institute of Technology in 1966 (10). This program interacts with a person at a terminal much like the interaction of a psychiatrist with a client. It was found that people where willing to share intimate details of their life with the computer. A typical session might go as follows.

```
(SPEAK UP)
    (I AM WORRIED ABOUT THE WEATHER)
(HOW LONG HAVE YOU BEEN WORRIED ABOUT THE WEATHER?)
    (SINCE MY MOTHER PASSED AWAY)
(TELL ME MORE ABOUT YOUR FAMILY)
    (THEY ARE ALL INTO COMPUTERS)
(DO MACHINES FRIGHTEN YOU)
    (NO)
(PLEASE DO NOT BE SO SHORT WITH ME)
    (I AM SORRY)
(EARLIER YOU TALKED OF YOUR MOTHER)
```

Despite ELIZA's seemingly insightful conversation, it is a "stupid" program. It has no real understanding of the user. It builds no model of the problems it seems to discuss, but depends instead on superficial key-word observations.

The first effort to deal with sytax, semantics, and reasoning ability in a completely integrated way, was made by Terry Winograd at the Massachusetts Institute of Technology (MIT) (11). His program was written to control a simulated robot manipulating simulated blocks on a simulated tabletop. Winograd's work dealt with image understanding and robotics (to be discussed later). To execute this task he wrote SHRDLU, an extensive interface of

language understanding. This interface converted words into program fragments, and sentence structure into an ordering of these fragments. Even with the integration of semantics and reasoning, SHRDLU is still not much more than a template matching system, not unlike ELIZA.

The most progressive work in this area has been pursued by those who argue that grammer, per se, is beside the point. They maintain that the real business of human language is not on the level of language itself but at a level of "concepts". Two noted scientists with this view are Roger Schank of Yale University and Marvin Minsky of MIT. Schank has written a program, FRUMP, that takes into account purpose and expectation through the use of "scripts" (12). These scripts have a typical sequence of actions; say, for instance, in a resturant. "Restaurant" may have various meanings for each of us depending on our culture and our upbringing. However, there are some universal connections we all make to a restaurant, e.g., food, server, cook, eating utensils, perhaps money, tables, chairs, music, etc. These universal "concepts" can be programmed into a computer as knowledge. It is a model of "conceptual dependency".

Minsky's work proceeds along this same vein. He employs the notion of a "frame," which is a method of representing a stereotyped situation (13). Attached to each frame are several kinds of information; e.g., how to use the frame, expectations in a frame, and what to do if the expectations are not met (defaults). For example, when someone hears "Jill threw up the

ball", s/he immediately conjures a secenario that reflects her/his knowledge of "Jill", "ball", "threw up", etc.. My image is one of a 5-6 year old girl tossing a large red rubber ball into the air. I know by the context of the word threw (even with jutaposed word "up") that Jill did not vomit, and the size and color of the ball comes from defaults I have aquired through time (even though the information is unspecified). This type of knowledge can be programmed into a computer through Minsky's frames. With incomplete information, the computer can still make inferences. If this notion of defaults is in fact how our minds operate, it might explain discrepancies in accident reports given by a group of people who all witness the same incident.

For both Minsky and Schank the answer to building a computer that can *understand* human language is to first understand how the human brain does it. One weakness in their work, as Barbara J. Grosz of SRI International points out is in the area of "speech acts" (14). For example a sentence such as "Can you pass the salt?" is not just a request for true-false information about the world. A robot who answers "yes" to such a question would be very annoying. The sentence is in fact an attempt to elicit action; the computer would have to understand that.

It is difficult to assess where this research will lead us. At best it is hoped that one day we will understand what *understanding* is. More realistically, we may simply learn to differentiate what can be symbolically processed (computers) from what will require some other model. It is certain, however, that

we will succeed at designing better ways to communicate with computers.

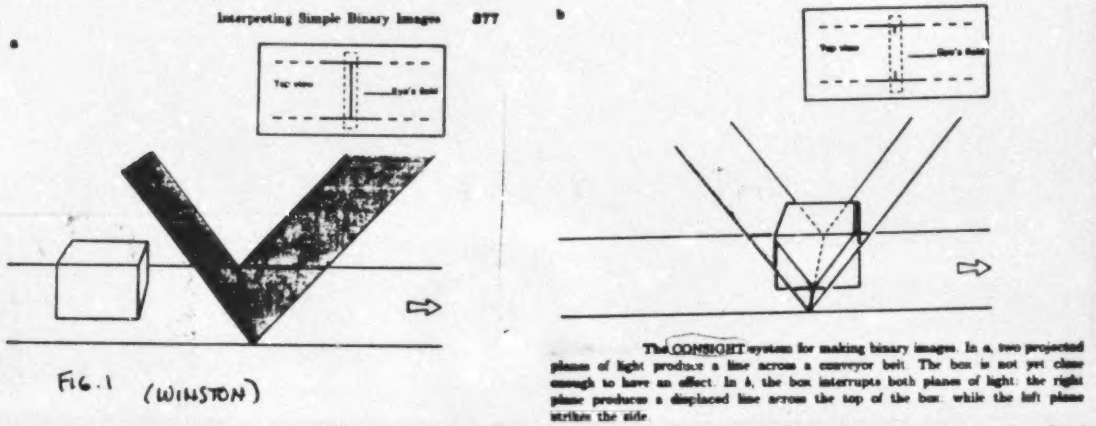
IMAGE UNDERSTANDING AND ROBOTICS

Computers have the potential to do work that people dislike--jobs that are dangerous, demeaning, hopelessly boring, or poorly rewarded. Jobs like these must be done by flexible robots or sophisticated armamentaria with very advanced techniques in artificial intelligence. Thus far, special-purpose machines with narrow domains dominate the market. Development of many capabilities such as reasoning about motion in space, dealing with force and touch information, and computer vision are needed. Image understanding, the study of the capability of a computer to see and properly interpret the 3-dimensional world, is very complicated and requires a great deal of knowledge to to fully understand it. Only a few specific techniques will be presented here to give the reader some idea of what has been done. First we will look at a proven application of the program, CONSIGHT. Then there will be a brief discussion on two particular visual processing problems that are better understood than most.

To familiarize you with an application of computer vision, we will examine CONSIGHT, a program that uses binary images to locate objects on a conveyor belt (1). The use of binary images is a shortcut way of "seeing". To get a binary image, each degree of brightness in an image is compared with a threshold value. If the brightness is above the threshold, the binary value is taken to be

BEST COPY AVAILABLE

one: otherwise it is zero. A system is said to use *structured light* whenever the system's operation depends critically on a special light arrangement. Rather than thresholding ordinary images, CONSIGHT uses the structured light arrangement shown in Fig. 1. CONSIGHT constructs the equivalent of a binary image of objects that pass through two planar sheets of light projected onto a moving conveyor belt at a 45 degree angle.



Interpreting Simple Binary Images 377

The CONSIGHT system for making binary images. In a, two projected planes of light produce a line across a conveyor belt. The box is not yet close enough to have an effect. In b, the box interrupts both planes of light: the right plane produces a displaced line across the top of the box, while the left plane strikes the side.

FIG. 1 (WINSTON)

When there is no object in view, the planar sheets make a bright line across the belt. When the planar sheets are obstructed by an object, the light is displaced. A sequence of belt positions produce a sort of silhouette of an object equivalent to a binary image (Fig. 2).

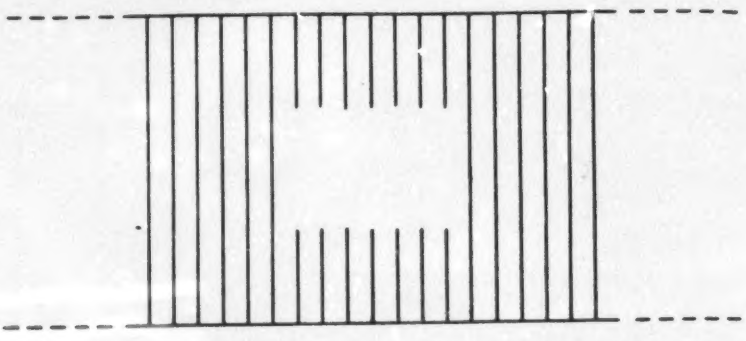


FIG. 2 (WINSTON)

Figure 10-33. The image of a box as seen by CONSIGHT. As a box moves past the camera's narrow field of view, gaps are produced in the line seen by the camera. These gaps constitute a negative image of the light-displacing box.

CONSIGHT cannot distinquish between parts amassed in a bin, but when indiviual items are separated on a conveyor belt, CONSIGHT enables robots to pick up the objects. Of course, evenually robots will need to do more sophisticated tasks involving manipulation and reasoning capabilites, as well as more advanced vision systems.

The first of the processing problems, the *binocular stereo* problem, is the notion of finding the distance from the viewer to a surface using two slightly different viewpoints. Stereo vision uses information from two eyes to determine distance. This method works because it is easy to determine the distance to an object once the object is seen in two images by eyes with known separation.

In Fig.3, we see from above an image point p, two lenses x and y, and two image planes. The axis of the lenses are parallel. Both lenses are separated from the image planes by their focal length f, and they are separated from each other by a baseline distance b. The point P is at a distance l from the left lens axis and r from the right. Similarly, P appears on the left image at a distance α from the left lens axis and in the right image at a distance β from the right lens axis. Solving the equation derived from the similar triangles we end up with the following expression for d.

$$d=fb/(\alpha+\beta)$$

Thus distance to a point is inversely proportional to $\alpha+\beta$, the amount of shift of the points' position in one image relative to

BEST COPY AVAILABLE

the points' position in the other. This shift is called disparity (1).

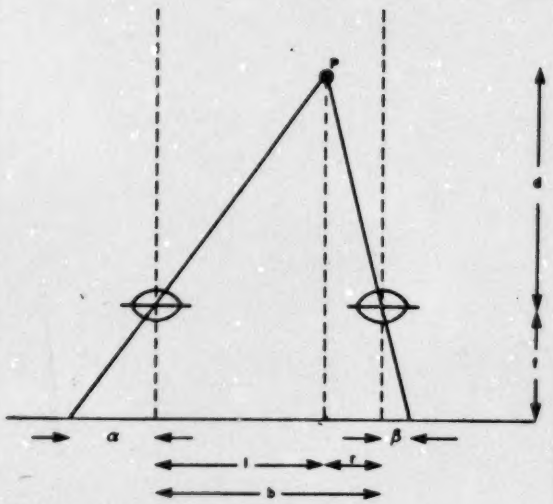


FIG. 3 (WINSTON)

The geometry of the two eyes involved in doing stereo. The distance to the object, d, is given by $d = fb/(\alpha + \beta)$.

The second problem, *shape-from-shading* , deals with determining surface direction at each point in an image by exploiting the way surfaces in the image are shaded. This leads to understanding the shape of the image. The amount of light one sees reflected from a surface depends on the material covering the surface and on the various angles shown in Fig. 4 .

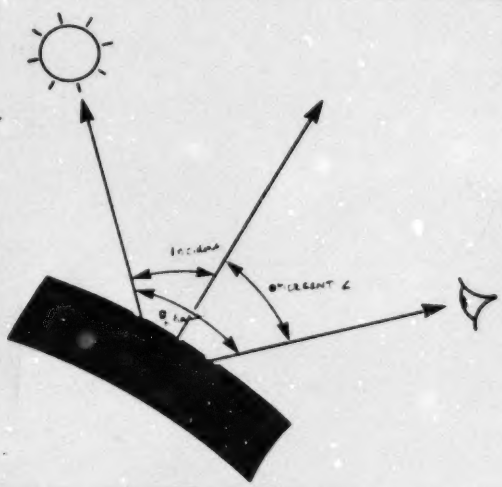


FIG. 4 (WINSTON)

In this illustration, we have a surface-centered view of the relative positions of the viewer and the light source and the relative orientation of the surface. The angles are e between the surface normal and the viewer; i between the surface normal and the light source; and g between the viewer and the light source. For many surfaces, observed brightness is proportional to $\cos i$ and independent of e and g.

The emergent angle *e* is between the angle that is normal (perpendicular) to the surface and the viewer direction; the incident angle *i* is between the surface normal and the light-source direction; and the phase angle *g* is between the viewer direction and the light-source direction. On a lambertain surface, one that looks equally bright from all viewpoints, the observed brightness depends only on the dirction to the light source. The formula reduces to E= cos *i* where is the surface *albedo*, *E*, is a constant for any particular surface material, and *i* is the incident angle (1). Once the surface direction is used to predict surface brightness, other techniques are employed to construct simulated images of the viewed object. The above processes are extremely sketchy and incomplete, they are presented here only to give the reader some idea of what problems are involved in developing a robot that can see. The above examples are only a fraction of what is being done in the field of image understanding. For more details refer to references.

LEARNING

The last subcategory is the area of study that considers the computer's ability to *learn* . It is generally believed that it is the ability of humans to learn that distinquishes them from the rest of the animal kingdom. It should then come as no surprise that this talent is the hardest of the intelligence-demonstrating talents to explain and to capture in A.I. procedures.

Researchers have concentrated their work on three primary approaches: acquisition of knowledge, search, and heuristics. Each will be dicussed here.

You might be asking "Where is the learning in the 'acquistion of knowledge', or 'search'?" It is true these two types of procedures are more akin to tools. The real learning comes when these tools are used in conjuction with a program that keeps track of the success/failure rate of certain paths taken and uses this knowledge the next time the same situation arises.

Emphasis on knowledge as a prerequisite to learning is important. The idea is that machines, like humans, will learn best when they learn slowly--when they relate each new concept to what is already known instead of trying to organize a whole mass of facts by some instantaneous gestalt. William A. Martin put it this way:

Martin's Law: You can't learn anything unless you almost know it already.

Clearly then any program that is expected to learn must have many banks of knowledge available for its base of reference. Also this program must be able to deduce new knowledge and continually add it to its base.

Search problems are ubiquitous, showing up everywhere A.I. researchers go. Many sophisticated search procedures have been developed. Included in this set of procedures are *depth-first search, hill climbing, breath-first search, beam search, best-first search, British Museum Procedure, branch and bound,*

discrete dynamic programming, *A*, and others. Often additional procedures are piggy backed with these search techniques to further the efficiency and/or effectiveness. A couple of these procedures are *mimimax* (a look ahead procedure) and *Alpha-Beta* (a procedure that prunes the potential possibilities in a given situation). One can see the importance of quick and accurate searches by looking at game playing, where the major work on leaning has been done.

The classic example is the game of chess. Any chess-playing program quickly has to face the fact that there are some 10^{120} sequences of legal moves in the game. A completely general program would have to examine them all. Yet the fastest computer built could spend a lifetime of the universe at it and still have a long way to go (15). Thus, searches developed in A.I. must use extensive networks and trees (a type of data structure) along with heuristics (rules of thumb) to solve problems.

Perhaps the procedure that more closely resembles the way in which we as humans learn is "induction heuristics". This procedure enables learning class descripition from positive and negative examples. With these heuristics a program can learn, for example, that an arch consists of one brick that must be supported by two others that don't touch each other. *Require-link*, and *forbid-link*, are two such heuristics. This procedure requires a cooperative teacher (programmer) who presents carefully chosen samples of arches and non-arches. The non-arches or *near-miss* samples must differ from the arch samples in only one small way

(remember Martin's law). Consider the sequence shown in Fig. 5. Note the series starts with a a typical arch (a). From this sample we want the learning procedure to gain a notion of what an arch is, ie. a structure that involves two standing bricks that together support a third brick lying on its side. Each subsequent sample should make a particular point. The configuration (b) is a *near-miss* and is intended to teach the program that the support links are important. In the second *near-miss* example (c), the two standing bricks are touching. Nothing else is significantly different from the original example. Therefore the procedure gains the knowledge that if a configuration is an arch, the supporting bricks must not touch. In the last configuration (d), the teacher could have presented it as an arch or a non-arch, depending on his/her discretion. If it is viewed as an arch, as it is here, then the learning program would learn that an arch could have as its supported unit a brick or a wedge. This broadens, rather than limits, the candidates in the arch classification. Conversely, if it was presented as a *near-miss* the class of arches would be more limited.

FIG. 5 (WINSTON)

A sequence of examples and near misses for learning about arches.

BEST COPY AVAILABLE

A data bank is built using the knowledge collected and the heuristics. For example, the required link established in the first *near-miss* would indicate a "must-support" link between each of the standing bricks and the supported brick, (Fig. 6) (the "left-of" link was established by the original sample). The second *near-miss* sets up a "must-not-touch", link between the two standing bricks (Fig.7). Other heuristics such as *climb-tree*, *enlarge set*, *drop-link*, and *close-interval* can also be used (3).

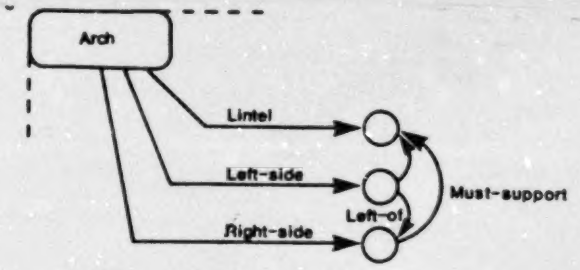


Fig The require-link generalization rule. Compared with the ARCH frame in *a*, the near-miss frame in *b* lacks SUPPORT links. The conclusion is that SUPPORT links are essential, so the SUPPORT links in the ARCH frame are altered, indicating that they are required in all arches, as shown in *c*. The LEFT-OF link is shown to emphasize the need for evidence that is sufficient to establish the correct correspondence between the parts of the arch and the parts of the near miss. Many links have been omitted from the drawing for clarity.

FIG.6
(WINSTON)

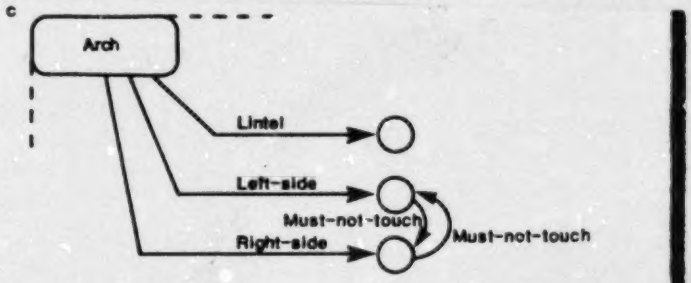


Fig The forbid-link generalization rule. Compared with the ARCH frame in *a*, the near-miss frame in *b* adds touch links. The conclusion is that the touch links must not be present, so touch links are added to the ARCH frame, altered to indicate that they are forbidden in all arches, as shown in *c*. Many links have been omitted from the drawing for clarity.

FIG.7
(WINSTON)

In conclusion, in induction heuristics, learning is a cooperative effort between a learner (computer) and a teacher (programmer). The teacher supplies samples and near-misses in a careful sequence, leaving the learner to determine exactly what to do after each.

If this descripion of "learning" has left you feeling bewildered, do not despair. As mentioned, learning in humans is not yet understood, and procedures to impliment learning in computers are even less understood. Learning is, however, an important step in utilizing the computer to its fullest capacity, and will probably continue to be an important area of study.

In the above sections on "expert systems," "natural language," "image understanding," and "learning," I have tried to give the reader a sense of what A.I. is. This is by no means a complete picture of all that is going on in the field. In the 1960's A.I. took a back seat in development due to economic problems, but has in recent years seen a restoration of interest. Government agencies and major companies are funneling large amounts of money and energy into developing the science and the engineering of Artificial Intelligence. Next we will take a close look at what A.I. developments have been made specifically in the field of education.

PART II. A SPECIFIC APPLICATION OF A.I. IN EDUCATION, WUMPUS ADVISOR

A new generation of Computer Aided Instruction (C.A.I) is evolving uses Artificial Intelligence techniques. These new programs, Artificial Intelligence Computer Aided Instruction (A.I.C.A.I) incorporate the concepts of Artificial Intelligence while maintaining the goals and objectives of Computer Aided Instruction. One of the goals of our educational system is to teach better ways to think about problems in general. It has been found that games can be an ideal teaching tool to obtain this goal. Most games represent some facet of life but in a much simpler domain, thereby allowing the student to develop problem solving abilities in an environment that is conducive to learning. These games can be fun and useful but the student often reaches a plateau in his/her game playing and then the learning process halts. A solution to this problem is to encourage the student to improve his/her game playing by offering suggestions for different game strategies. It is possible for a computer to monitor the progress of the game and offer suggestions when warranted. This is the intent behind WUSOR, a program written to advise the user in the game of WUMPUS (sometimes called WUMPUS HUNTER) (16).

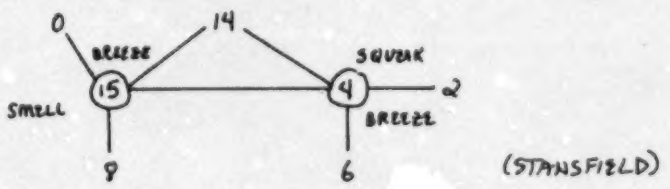
WUMPUS is a maze-exploration game that encourages deductive processes and develops a sense of probabilities. The player must seek out and kill a monster, the Wumpus. The Wumpus warren consists of a number of caves (approximately 20) connected by

tunnels. Initially the player only knows the structure of his/her immediate environment. S/he knows the number of the cave s/he is in and of all the caves directly connected to it by tunnels. Every time s/he makes a move, which must be into a neighboring cave, s/he is told the cave-numbers neighboring his/her new cave. The dangers of the warren are pits (usually 3), bats (also, usually 3), and the Wumpus. They, like the "hunter", are initially randomly placed in the warren. Entry into a cave with the Wumpus or a pit is instant death. If the player moves into a cave with bats, s/he is carried away by bats and dropped into a random cave. There are clues which help avoid these hazzards. The player hears a squeak if s/he is one cave away from a bat, and s/he can feel a breeze if s/he is one away from a pit. S/he can smell the stench of the Wumpus from up to two caves away but cannot tell the distance directly. The "hunter" has a bow and five arrows which s/he can fire at any time into a neighboring cave. The arrow will ricochet through the warren at random for a distance of up to five caves.

A scenario of the game might proceed as follows (17).

WUMPUS (W): You are at 15 with neighbors 4, 14, 0, and 8.
Brrrr! I feel a draft. You are near a pit.
What a stench. I smell the Wumpus! What now?

Player (P): 4



(Circled caves are those [illegible]ed)

(W): I wish you luck. Your are at cave 4 with neighbors 14, 6, 2, and 15. Brrrr! Squeak! A bat is near. What now?

(P): 14

WUSOR (advisor)

(A): This is a dangerous move. You have been in two neighbors of 14 and you felt a draft in both. This is double evidence so it is probable that a pit is there. What now?

(P): Advise

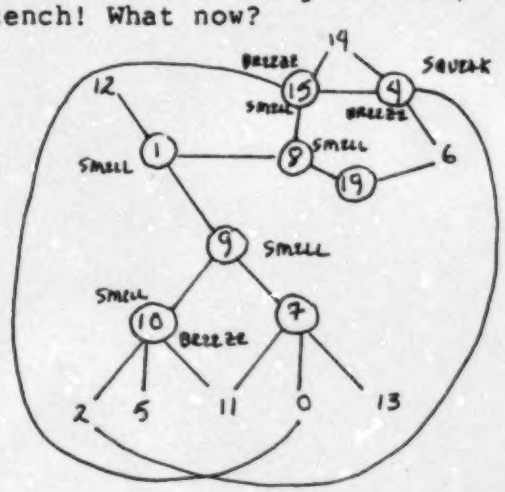
(A): Cave 8 is a better cave to explore What now?

(P): Why?

(A): There is no danger from the Wumpus at 14 or 8. It is possible 8 contains a pit, but it is more likely that there is one at 14. What now?

. . .

(W): You are at cave 1 with neighbors 12, 8, and 9. What a stench! What now?



(STANSFIELD)

(P): 12

(A): That is a fatal move. Would you like advice?

(P): Yes

(A): The Wumpus is there by the following argument:

Cave 8 must be exactly two away from the Wumpus

since it smells and cave 19 doesn't. This means that either cave 1, 15, or 9 must be next to him.

Cave 15 is two away since it is next to 4 which doesn't smell. Cave 9 isn't one away since you visited cave 7 and that didn't smell. Therefore, by elimination , cave 1 is one away. Hence, one of its neighbors must be the Wumpus.

The neighbors of 1 are 8, 9, and 12. You have visited 8 and 9 so, by elimination, the Wumpus is at 12.

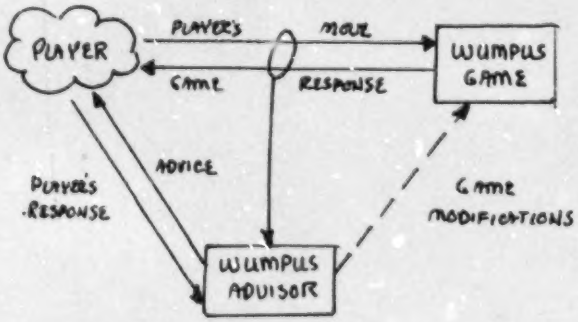
SHOOT

Which cave would you like to shoot into?

(P): 12

(W): Congratulations you have shot the Wumpus!

Figure 8 shows the interaction format between WUMPUS, WUSOR, and the player.



(CARR)

The WUMPUS advisor, WUSOR, applies many of the techniques developed in A.I.. It centers around a rule based Expert which uses heuristics to arrive at results that are both reliable and understandable to humans. The students' responses are then compared to those generated by the Expert. The rules are organized into a curriculum which is taught to the player by the

WUMPUS advisor. This organization allows selective advice and appropriate presentation of rules justifying the advice, appropriate to the player's level of development. The following are some examples of the WUSOR production rules and heuristics.

RULE 1: Goal--to prove a cave is safe from the Wumpus.
A cave is safe:
a) If it has been safely visited
b) If an arrow has been fired into the cave and no Wumpus was hit
c) If there is a NO-SMELL cave within two caves of it

RULE 4: Goal--to prove a cave is one-away.
A cave is one away from the Wumpus if it has a neighbor which is two away and all other neighbors of that cave are more-than-one-away.

LOGIC RULES:
L0: A cave can be marked as "zero away" if it was visited and found to contain danger
L4: If a cave is visited and there is a warning, then that cave is "less than (N+1) away".

HEURISTICS:
C0: Safe unvisited caves are preferable to all others
C1: Given that it is necessary to take some risk, those moves with the lower risks for all dangers are preferable
C6: When the player is in a very dangerous section of the warren with no known route to a safer area, it is sometimes advisable to seek out bats in the hope that they will carry the player to a safer section of the warren

The advisor, WUSOR, is able to give information about where the player is located in the warren, where s/he has been, and what was learned at each cave that was visited. It also can give advise based on the heuristics. This information is communicated to the player through "English Generation Routines" (17). Each of the routines/rules of the Expert has a corresponding English

routine/rule. When an Expert rule is used to evaluate a warren state, its sister English rule can be generated to inform the player of the reasoning behind the given advice. The translation from Expert rule to English rule is not word for word. Consideration is given to the idiosyncracies of the English language. For example, the English routines do not say "zero caves away from", or "less than (N+1) caves away" (see LOGIC RULES, page 25), but instead say "next to" and "no more than 1 (or 2 etc.) caves away".

The English routines also delete detail, leaving a concise, quickly readable summary of the rules used by the Expert. Not only do the English routines simplify explanations, but with the use of three other routines, "Mover Comparer," "Psychologist," and "Student Model," WUSOR is able to determine what moves can (or should) be explained to the student. This is accomplished by ascertaining which rules are already known to the player. To determine what these rules are, WUSOR keeps track of and evaluates information such as the following.

1) How often the student seems to have applied the rule
2) How often the student has had the rule explained to him/her
3) The student's knowledge of other rules (upon which this rule depends)
4) How quickly the student learns
5) The student's initial knowledge of the rules
6) The time period that has elapsed and how forgetful the student is.

Using these results, WUSOR is able to incorporate student modeling.

WUSOR, with its many parts (Advisor, Expert, Mover Comparer, Psychologist, and Student Model), is a successful A.I.C.A.I. program. It incorporates many A.I. techniques to go beyond the standard C.A.I. limitations.

PART III: SUMMARY

Many C.A.I. teaching programs are "fact dispensing" machines (14). They use the "empty bucket" learing theory that views each of us as an empty receptical waiting to be filled with facts. *Process*, the key element of learning, is not incorporated into this system. Rather, the student should learn a generalized procedure that will generate an understanding of a range of situations. In a sense, the student is being programmed. By changing the paradigm from facts to procedures, the whole learning process becomes greatly enriched.

C.A.I. programs need models of situations and pupils if they are to *understand* what is going on and act appropriately. By studying simple teaching situations and modeling teaching programs after them, we gain insight into the processes underlying learning and teaching. Computer programs need to be provided with practical procedures for making decisions about teaching. Precisely formulated knowledge of their subject matter is required for proper interpretation, modeling and action within a variety of teaching situations. The programs also need an expressive means of communication such as natural language both for interpreting

students' behavior and making effective responses.

A good A.I.C.A.I. program will teach the student how to think. It will compare the learning task with the student behavior. Particular attention should be given to things the pupil does wrong, forgets to do, does unnecessarily, or does in the wrong order. Computer Science is rich in techniques to accomplish these goals and a has a vocabulary of precise concepts for describing these processes. As we have discussed, there are formulated concepts in A.I. concerning perception, natural reasoning, organizing knowledge, planning, and so forth; these can be used for new descriptions of the learning and teaching process. With the incorporation of Artificial Intelligence into the field of Computer Aided Instruction, the future of computers in the educational system is a bright one.

REFERNECES CITED

1) Winston,P.H., "Artificial Intelligence," Addison-Wesley Publishing Company, Inc., 1984.

2) McDermott, J., "R1:A Rule Based Configurer of Computer Systems," Artificial Intelligence, vol.19, no.1, 1982.

3) Shortliffe, E.H., "MYCIN: Computer Based Medical Consultations," Elsevier, New York, 1976. Based on a Ph.D. thesis, Stanford University, Stanford, CA, 1974.

4) Mylopoulos, J., "An Overview of Knowledge Representation," Proc. Workshop Data Abstraction, Databases, and Conceptual Modeling, June 1980.

5) Nilsson, N.J., "Principles of Artificial Intelligence," Tioga, Palo Alto, CA, 1980.

6) Huffman, D., "Impossible Objects as Nonsense Sentences," Machine Intelligence 6, D. Meltger and D. Michie, eds., Edinburgh University Press, 1971, pp. 295-323.

7) Clowes, M., "On Seeing Things," Artificial Intelligence, vol. 2, no. 1, 1971, pp. 79-116.

8) Newell, A., and Simon, H.A., "GPS, a Program that Simulates Human Thought," Computers and Thought, E.A. Feigenbaum and J.A. Feldman, eds., McGraw-Hill, New York, 1963.

9) Fikes, R.E., and Nilsson, N.J., "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving," Artificial Intelligence, vol. 2, no. 3, 1971, pp. 189-208.

10) winston, P.H., "Artificial Intelligence," (First edition), Addison-Wesley Publishing Company, Inc.,1977.

11) Winograd, T., "Frame Representations and the Declarative Procedural Controversy," Representation and Understanding, Edited by Daniel G.Bobrow and Allen Collins, Academic Press, New York, 1975

12) Schank, R.C., "Conceptual Dependency: A Theory of Natural Language Understanding," Cognitive Psychology, vol. 3, no. 4, 1972.

13) Minsky, M., "A Framework for Representing Knowledge," Psycology of Computer Vision, McGraw-Hill Book Company, New York, 1975.

14) Grosz, B.J., "Natural Language Processing," Artificial Intelligence, vol. 19, no. 2, 1982.

15) Waldrop, M., Science, Vol. 223, March 23, 1984.

16) Stansfield, J.L., Carr, B.P., and Goldstein, I.P., "WUMPUS Advisor I, A First Implementation of a Program That Tutors Logical and Probabilistic Reasoning Skills," Artificial Intelligence Laboratory of MIT, Cambridge, Mass., Oct. 1976.

17) Carr, B., "WUSOR II," Artificial Intelligence Lab of MIT, Cambridge, Mass., May 1977.

SUPPORTING LITERATURE

1) Edilson, E., "Programmed to Think," MOSAIC, vol. 11, no.5 Sep./Oct. 1980, pp. 19-23.

2) Cromie, W.J., and Edson, L., "Before They Can Speak, They Must Know," MOSAIC, vol. 15, no. 1, 1984, pp. 29-35.

3) Whaland, N., "When is a Program Intelligent?" Creative Computing, vol. 7, no. 2, Feb. 1981, pp. 44-49.

4) Shortliffe, E., "MYCIN: Computer Based Medical Consultations, 1976.

5) Tilmann, M., "MUSHROOM: An Expert System," Michigan State University, E. Lansing, MI, June 1984.

6) Davis, R., Buchanan, B., and Shortliffe, E., "Production Rules as a Representation for Knowledge-Based Consultation Program," Artificial Intelligence 8, North-Holland Publishing Company, Stanford, Cal., 1977.

7) Kendig, F., "A Conversation With Roger Schank," Psychology Today, April 1983, pp. 28-36.

8) Schechter, B., "The Science of Deduction," Discovery, Jan. 1982. Jan. 1982, pp. 26,27.

9) Waldrop, M.M., "The Necessity of Knowledge," Science, Vol. 223, March 1984, pp. 1276-1282.

10) Waldrop, M.M., "Artificial Intelligence (I): Into the World," Science, vol. 223, Feb. 24, 1984, pp. 802-805.

11) Feigenbaum, E.A., "The Building Blocks," IEEE Spectrum, Nov. 1983, pp. 77-91.

12) Nau, D.S., "Expert Computer Systems," IEEE, Feb. 1983.